ARMY RESEARCH LABORATORY

# Analysis of Cisco Open Network Environment (ONE) OpenFlow Controller Implementation

**by Curtis Tade, Venkat Dasari, and Vinod K. Mishra**

**ARL-TR-7020** **August 2014**

## NOTICES

### Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

# Army Research Laboratory
Aberdeen Proving Ground, MD 21005-5067

**ARL-TR-7020** **August 2014**


# Analysis of Cisco Open Network Environment ONE OpenFlow Controller Implementation


**Curtis Tade**
**University of Texas at Austin**

**Venkat Dasari**
**Secure Mission Solutions**

**Vinod K. Mishra**
**Computational and Information Sciences Directorate, ARL**

| REPORT DOCUMENTATION PAGE | Form Approved OMB No. 0704-0188 |
|---|---|



| 1. REPORT DATE *(DD-MM-YYYY)* | 2. REPORT TYPE | 3. DATES COVERED (From - To) |
|---|---|---|
| August 2014 | Final | June–August 2013 |

| 4. TITLE AND SUBTITLE | 5a. CONTRACT NUMBER |
|---|---|
| Analysis of Cisco Open Network Environment (ONE) OpenFlow Controller Implementation | 5b. GRANT NUMBER |
| | 5c. PROGRAM ELEMENT NUMBER |

| 6. AUTHOR(S) | 5d. PROJECT NUMBER |
|---|---|
| Curtis Tade, Venkat Dasari, and Vinod K. Mishra | 5e. TASK NUMBER |
| | 5f. WORK UNIT NUMBER |

| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) | 8. PERFORMING ORGANIZATION REPORT NUMBER |
|---|---|
| U.S. Army Research Laboratory<br>ATTN: RDRL-CIH-N<br>Aberdeen Proving Ground, MD 21005-5067 | ARL-TR-7020 |

| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) | 10. SPONSOR/MONITOR'S ACRONYM(S) |
|---|---|
| | 11. SPONSOR/MONITOR'S REPORT NUMBER(S) |

**12. DISTRIBUTION/AVAILABILITY STATEMENT**

Approved for public release; distribution is unlimited.

**13. SUPPLEMENTARY NOTES**

**14. ABSTRACT**

The benefits of Software-Defined Networking (SDN), when fully realized, offer many improvements over the current rigid and complex networking paradigm. By virtue of separating the control plane from the forwarding devices and handing its control over to a centralized external device, SDN removes many obstacles to networking innovation and research. OpenFlow protocol is a key enabler of SDN and provides a centralized control channel for effective state management. Simplified and centralized state management using SDN is highly desirable for rapid and secure network deployment in the battlefield. As the open-source community continues to progress with more stable software, the vendor support and implementation are leading the way in network innovation. Cisco has recently joined the list of SDN vendor supporters with the debut of its Open Network Environment (ONE) controller in 2013. Cisco ONE is built to OpenFlow 1.0 specifications and is said to be fully compliant with standard OpenFlow. In addition to the standard OpenFlow implementation, Cisco has also introduced a control plane in their switches as backup for the situation in which the external controller becomes unavailable. The U.S. Army Research Laboratory has obtained a privileged copy of the Cisco ONE controller from Cisco. As part of a summer project, the Cisco ONE controller has been analyzed and compared with the standard OpenFlow Floodlight controller for features and functionalities like handshake, connection setup, switch management, and security.

**15. SUBJECT TERMS**

OpenFlow, software-defined networking, Cisco ONE, SDN controller

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT | b. ABSTRACT | c. THIS PAGE | | | Vinod Mishra |
| Unclassified | Unclassified | Unclassified | UU | 18 | 19b. TELEPHONE NUMBER (*Include area code*)<br>(410) 278-0114 |

## Contents

## List of Figures



## List of Tables

# 1. Introduction

The development of current network architecture has remained relatively stagnant since its inception, and its architecture is largely based around the decentralized control of networking devices through an embedded control plane in each device. Proprietary control plane, with its closed application programming interface (API) and hidden data plane, has become a great hurdle in innovating packet-forwarding technologies. Network device roles are strictly defined with little or no flexibility. In Software-Defined Networks (SDNs), new ideas were engineered to overcome the limitations of traditional network technologies to fuel the stalled network innovations and promote their rapid deployment

SDNs are based on the idea of a centralized control plane that is separate from the network-forwarding plane (*1*). SDN essentially provides the paradigm for the control of how data (flow) moves through the network. Thus in order for SDN to function, a new protocol must be used to permit communication between the abstracted control plane and the remaining "dumb" forwarding network devices.

The Open Networking Foundation (the organization promoting the adoption of SDN through the development of open standards) introduced the OpenFlow* Standard based on the OpenFlow specifications released in early 2008 (*2*). OpenFlow is standards-based networking protocol that separates and externally centralizes the control plane. Working in conjunction with an open API, OpenFlow allows the user to interface with the controller and provides remote programming of the forwarding plane of OpenFlow-compliant network devices. At the application layer, the user provides input for network configuration (rules, processes) that interfaces with the controller via OpenFlow. Forwarding tables that define "flows" are then programmed (an algorithm is run to determine forwarding table layouts) and dispatched into all network devices (network topology is built in memory). Figure 1 shows a top-down view of the interaction between the user, the controller, and the network.

In the northbound API, the controller provides information gathered from the network and makes it accessible to the user. The user runs applications to interface with the controller through an open API, allowing the user input to determine network configurations for the controller. In the southbound API, the controller makes requests and receives responses from the switch (e.g., features of the switch, configuration parameters). The controller may also send modify-state messages to add/delete flows or read-state messages to collect statistics from the switch.

*OpenFlow is a trademark of Stanford University.

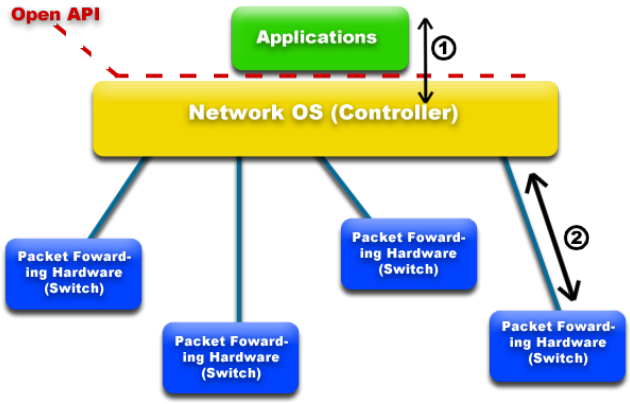



Figure 1. Overview of OpenFlow architecture.

The controller is the key SDN component in an OpenFlow-enabled SDN. It is the centralized abstracted control software that monitors and manipulates the network traffic flows. The controller provides the link between northbound and southbound APIs, allowing communication between network devices and end-user applications. The goal of this report is to provide an empirical analysis of the Cisco OpenFlow controller implementation. This is accomplished by comparing the Cisco controller to an existing open-source controller.

## 2. Cisco ONE Controller

Announced at Cisco Live in June 2012, the Cisco Open Network Environment (ONE) controller embraces the openness of SDN while maintaining proprietary Cisco features. Although the movement toward SDN is discouraging the necessity for proprietary switching technology, Cisco, possibly attempting to prevent some form of network commoditization, provides much of its own proprietary software for its SDN implementation. Seeing OpenFlow as more of benefit to academia and research-community than to data centers with “broadly defined network functionality” (*3*), Cisco has added deeper internals in their operating systems, plus hardware and ASICs, that can be accessed to extend and improve the network (*4*). See figure 2 for Cisco’s SDN model.

The Cisco SDN model embraces OpenFlow and extends it by adding their own extensions. Transport network services are tightly integrated into the packet-forwarding and control functions. Cisco’s OpenFlow controller, Cisco ONE, is built upon the source code from Open DayLight controller, whose development is being funded by IBM and Cisco. Cisco’s SDN model will also provide a research programmable interface for manipulating data plane and control plane communications (*3*). Cisco ONE supports both hybrid and native OpenFlow switches. It also opens up its northbound communications to application, which allows the applications to become aware of the network states for making intelligent decisions for moving data across an SDN network.

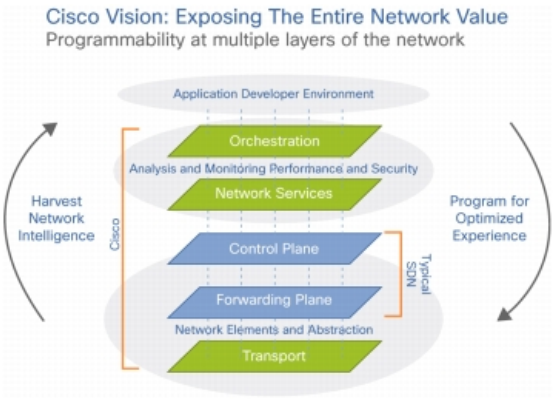



Figure 2. Cisco's vision for ONE (reproduced with permission by Cisco) (*3*).

In addition to a centralized and abstracted external control plane inside a controller, Cisco has decided to keep a backup control plane inside the switch to become active when a switch loses its connectivity to the external controller. This design modification is a bit controversial and goes against the standard SDN design principles, where the switches are just high-speed traffic-forwarding devices without any control plane. Whether or not this decision will close off Cisco from the anticipated multivendor integrated networks or provide its current user base with a reassuring familiarity of a Cisco-based SDN is yet to be determined. However, what is of greater importance is how the ONE controller performs in comparison to a standard reference controller (POX, NOX, Beacon, etc.).

## 3. Standard Reference Controller

For the purpose of this report, the Floodlight SDN controller was used as a standard for comparison with Cisco's ONE controller. Floodlight originated from the Beacon SDN controller. Beacon is a cross-platform Java-based OpenFlow controller created by David Erickson at Stanford University (*5*). Before being licensed under GNU General Public License (GPL version 2), a branch of Beacon was used to create Floodlight controller. Since then, the Floodlight Open SDN controller has become an enterprise-class Apache-licensed OpenFlow controller with wide support from developer communities and engineers at Big Switch Networks (*6*). Similar to the ONE controller, Floodlight is written in the Java programming language and is one of the few open-source controllers with a working graphical user interface (GUI), making it an ideal reference controller for the Cisco ONE.

## 4. Experiment/Calculations

The experimental test bed contains multiple hardware devices, including a host server and an OpenFlow-capable switch. Figure 3 clearly illustrates the experimental topology and its components.

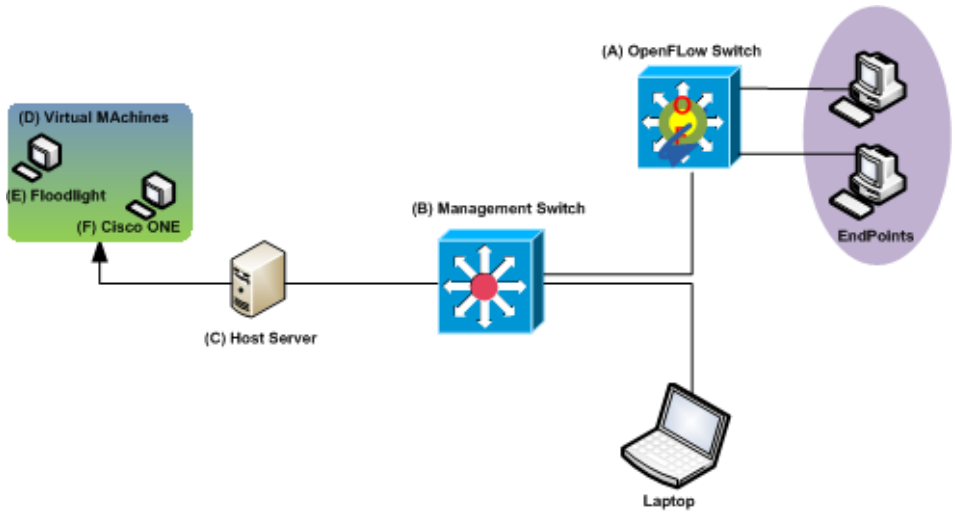



Figure 3. Experimental network setup topology.

As seen in figure 3, all devices are wired through a Cisco Catalyst 2960G-8TC-L data switch (B). The host server (C) is a Dell PC running Ubuntu, version 13.0, and hosts virtual machines (D) for the Floodlight (E) and Cisco ONE (F) controllers. The controllers are operated independently over a virtual bridge on the host server that connects through the physical link to the HP 6600-24G-4XG OpenFlow switch (A). Each link in the topology has an available bandwidth of 1 Gigabits per second (Gbps).

The topology in figure 3 was used to conduct experiments of the OpenFlow control channel setup, flow modification behavior, performance and stability, security of the controller, and overall features of the controller provided to the user.

## 5. Results and Discussion

### 5.1 Interface

The most obvious difference between the ONE controller and the Floodlight is that ONE has a built-in Web-based graphical interface, which is highly functional (figure 4), whereas Floodlight's Web applet is limited, and an additional application, Avior, is needed to supply

similar functionality. Both controllers succeed in providing a simple interface that provides the network administrator with an option to configure and manage the OpenFlow network topology in addition to using scripts and Command Line Interface. The ONE GUI is more feature rich with port recognition (to prevent assigning actions or match criteria to ports that are not connected) as well as the ability to install and uninstall a flow from switches without deleting it entirely. One of the major advantages of the ONE is the ability to remove flows after time-out (set dynamic flows). This feature should be common among all OpenFlow controllers; however, Floodlight requires the user to manually edit the staticflowpusher code's ParseRow() to allow for this.

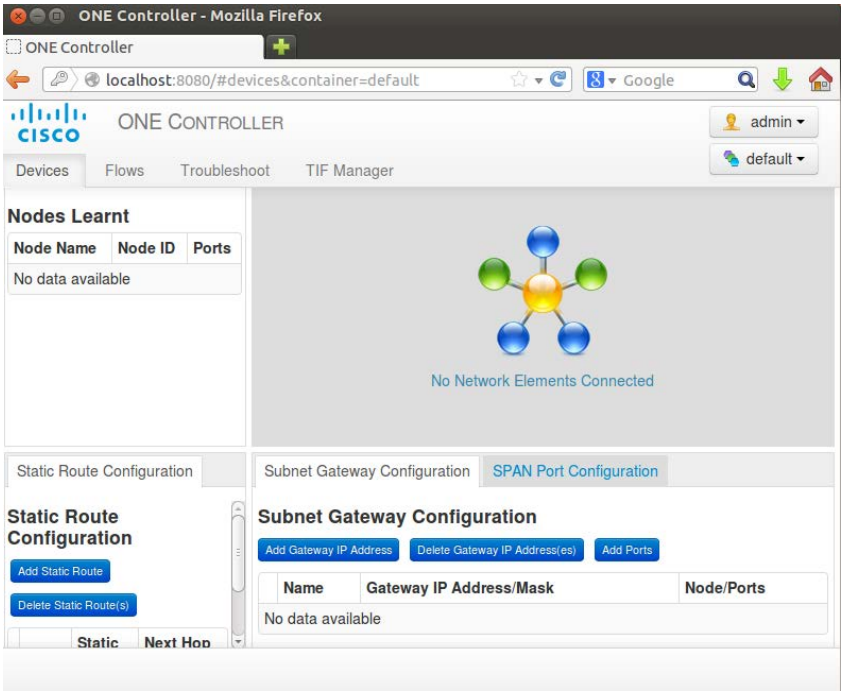



Figure 4. Cisco ONE GUI.

As shown in table 1, Cisco ONE's user interface is more mature, functionally comprehensive, and stable. It draws its strength from the experience learned from Open DayLight controller.

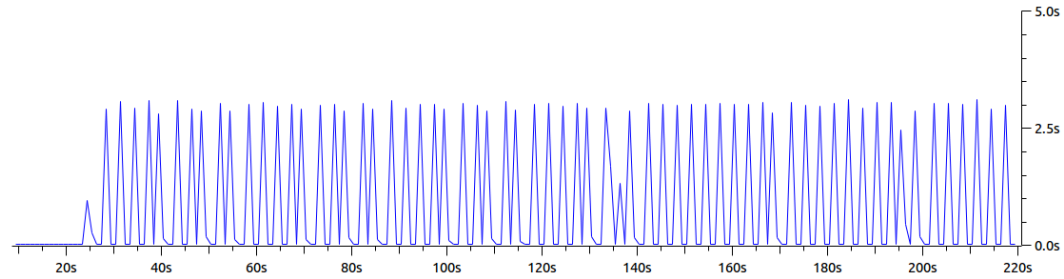
Table 1. Interface comparison.

| Feature | Floodlight | ONE |
|---|---|---|
| GUI | Requires Avior | Built-in Web applet |
| Port recognition | Not explicit | Yes |
| Install/uninstall flows | NA | Yes |
| Set dynamic flows | Requires code manipulation | Yes |

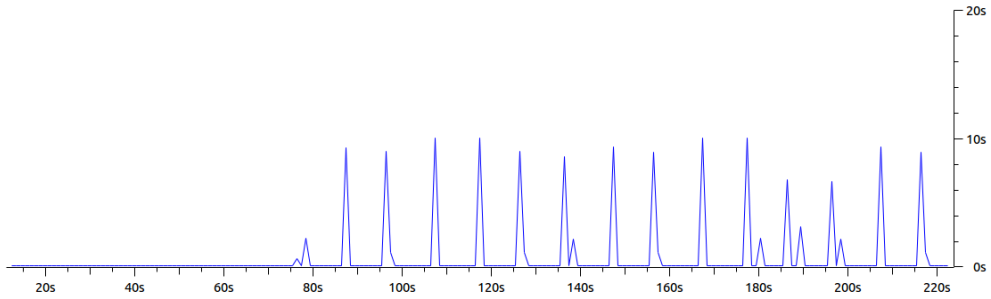
## 5.2 OpenFlow Control Channel Setup

The initial handshake between the controller and switch was monitored using packet analysis tools capable of understanding OpenFlow protocol. For this purpose, a new Wireshark application was compiled from its source by including support for OpenFlow as described by CPqD developers (*7*). Otherwise Wireshark does not understand the OpenFlow protocols for decoding the communications between the ONE controller and the switches.

Once the OpenFlow channel is set up between the switch and the controller, the switch states are constantly polled by the controller. In figures 5a and 5b, the initial handshake is shown as the response time between frames. The x-axis is the real time for wire capture, and the y-axis represents the time difference between the controller-switch communications. The peaks represent delays between the current frame and the last frame sent, where the initial peak is the handshake. Most peaks represent a Stats or Echo request by the controller with an almost immediate response from the switch.




(a)




(b)

Figure 5. (a) Floodlight handshake time delays and (b) ONE handshake time delays.

Worth noting is the frequency (chattiness) of the Floodlight controller communications compared to ONE. The higher frequency of frames (3-s delay compared to 10-s delay) is due to the persistent stats requests required by the Avior GUI to maintain an updated view of the network topology (table 2).

Table 2. Control channel setup.

| Feature | Floodlight | ONE |
|---|---|---|
| Stats requests frequency | High frequency ($\Delta t = 3$ s) | Low frequency ($\Delta t = 10$ s) |

These Read State Messages are used by the controller to query the switch for various state information elements (description of the OpenFlow switch, individual flow statistics, port statistics, etc.). Frequent state polling allows the controller to quickly learn and react to state changes in a switch and to update the topology before they can cause problems to higher-level protocols and applications requiring consistent link state for reliable performance. The extra frames may minimize a stale or split state, and the constant stat requests prove not to be a hindrance on the network in high-bandwidth network environments. However, in low-bandwidth environments, Cisco's infrequent state verification (and that of Floodlight without running Avior) may be more appropriate. Additional studies are required to understand the relationship between state synchronization dynamics and polling frequency in larger network topologies.

### 5.3 Control Channel Stability

The robustness of the control channel in the presence of high traffic loads on the network was evaluated by monitoring the controller/switch communications while flooding the network with high-speed traffic. Transmission Control Protocol (TCP) traffic was generated on the laptop and sent over the data switch to the host server at an average of 107 MB/s using the netcat utility.

The result for both controllers was similar. The Differentiated Service Code Point markings remained at the default state (000 or best-effort IP precedence), and while there were a large number of packet drops between the laptop and the host, the number of packet drops (OpenFlow traffic) from controller to switch was unaffected. This result would suggest high stability for both ONE and Floodlight with a possible explanation being that the control plane does not require high bandwidth for controller/switch communications.

### 5.4 Flow Modification Behavior

Flow modification is an essential function through which a controller will manipulate the traffic passing through a switch. The barrier requests and replies are essential to keep the network state consistent in forwarding devices in a network.

Several tests were conducted where flows were modified (added and deleted) for packets being sent matching input and output across ports 23 and 24 (cross-connected ports) on the HP switch.

Packet analysis shows that the ONE controller ensures the completion for each flow modification by sending "barrier requests to" and receiving "barrier replies from" the switch. Floodlight does not exhibit this behavior when sending flow modifications in Avior (table 3). Further observation shows that during the initial handshake, the ONE controller sends an initial flow mod (figure 6) to remove all previous flow data remaining on the switch. A switch has a finite number of flows that can be saved in the Ternary Content-Addressable Memory and cannot add new flows until

previous flows are removed. Thus, this action prevents operational problems when a switch with previously installed flows connects to the controller. The Floodlight controller does not explicitly exhibit this same function. Additional testing was done to determine the throughput (flows/second) of the controller using cbench; however, using cbench in the ONE controller to generate flows was not successful.

Table 3. Flow modification behavior.

| Feature | Floodlight | ONE |
|---|---|---|
| Modification verification | No | Barrier requests/replies |
| Initial flow mod | Not explicit | Mod delete after handshake |

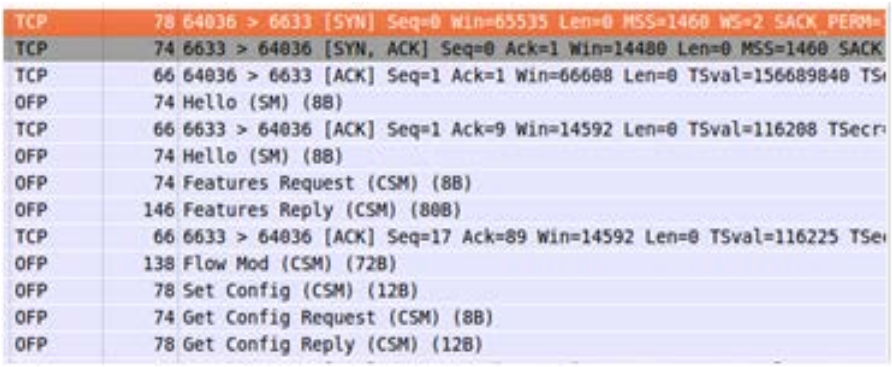



Figure 6. OpenFlow handshake followed by flow mod.

### 5.5 Controller Security

Using the proposed topology outlined in figure 3, basic security tests were performed on each controller using the laptop as the attacker.

Layer 2 Address Resolution Protocol (ARP) poisoning allowed for a successful MITM (man in the middle) attack on both the Cisco ONE and Floodlight controllers. DOS (denial of service) attacks were also successful on both controllers (table 4). The ability to penetrate/disrupt the controller/switch communications can be addressed by the vulnerabilities in the OpenFlow protocol and may not be specific to controllers. The OpenFlow Switch Specification suggests that “the switch and controller may communicate through a Transport Layer Security (TLS) connection.” However, by default, OpenFlow channel communications were in clear text and vulnerable to attacks like ARP poisoning. The TLS option should be the default OpenFlow channel setup option, and code needs to be modified to expose the hidden TLS options and make them the default. Additionally, the OpenFlow specification recommends using “alternative security measures” when communicating using plain TCP to revent eavesdropping, controller impersonation or other attacks on the OpenFlow channel. OpenFlow control channel communications do not circumvent general network-related problems and still depend on security measures applied to protect non-OpenFlow communications. But members of the research community are currently working to integrate security services into controller software.

Table 4. Controller security.

| Security Attack | Floodlight | ONE |
|---|---|---|
| MITM | Vulnerable | Vulnerable |
| DoS | Vulnerable | Vulnerable |

## 6. Conclusions

Overall, the Cisco ONE controller proves to be a competitive addition to existing OpenFlow controllers. Although the Cisco ONE is compliant with the OpenFlow 1.0 standard specifications, it maintains disparate design differences, as stated previously, from open-source OpenFlow controllers. The intended primary use of the ONE controller is pairing it with Cisco OpenFlow switching hardware, but the results of this experiment show that it can still be used effectively on non-vendor-specific hardware. ONE excels with a well-constructed user interface, inherited from Open DayLight controller, which has yet to be fully implemented by its competitors in the field. Additional features benefitted from using ONE with Cisco hardware could not be addressed by this study. The experiments concluded by this report are not exhaustive, and further testing will be continued. Some planned future work will include:

- Expansion of the test bed to a greater number of devices as well as more comprehensive flow modification experiments.
- Use of OpenFlow analysis tool cbench in conjunction with ONE controller.
- Use of additional reference controllers for comparative analysis.

## 7. References


1. Gringeri, S.; Bitar, N.; Xia, T. J. Verizon Laboratories. Extending Software Defined Network Principles to Include Optical Transport. *IEEE Communications Magazine* **March 2013**.
2. Open Networking Foundation. ONF Overview. https://www.opennetworking.org/about/onf-overview (accessed July 2013).
3. White Paper: OpenDaylight - An Open Source Community and Meritocracy for Software-Defined Networking, 2 April 2013.
4. Cisco Open Network Environment: Network Programmability and Virtual Network Overlays. http://www.cisco.com/en/US/prod/collateral/iosswrel/content/white_paper_c11-707978.html (accessed August 2014).
5. Erickson, D. OpenFlow at Stanford: Beacon Home. https://openflow.stanford.edu/display/Beacon/Home, 4 February 2013 (accessed August 2014).
6. Project Floodlight. Floodlight. http://www.projectfloodlight.org/floodlight/ (accessed August 2014).
7. GitHub, Inc., Web site. https://github.com/CPqD/ofdissector (accessed July 2013).

1 (PDF) DEFENSE TECHNICAL INFORMATION CTR DTIC OCA

2 (PDF) DIRECTOR US ARMY RESEARCH LAB RDRL CIO LL IMAL HRA MAIL & RECORDS MGMT

1 (PDF) GOVT PRINTG OFC A MALHOTRA

1 (PDF) DIR USARL RDRL CIH N V MISHRA

INTENTIONALLY LEFT BLANK.